In the Specification:

Please replace the paragraph beginning at page 2, line 9, with the following rewritten paragraph:

Another of the stages is a data transfer stage 12 during which the master apparatus A sends a message IN signaling that it is awaiting reception of the data read following the command. Not being available (ready), the microcontroller 24 28 of the slave apparatus sends a message NAK signaling a non-acceptance. At a later stage, the master apparatus A resends the message IN, to which the slave apparatus then responds by sending the read data DATA. The master apparatus then returns an acknowledgement of receipt ACK.

Please replace the paragraph beginning at page 3, line 27, with the following rewritten paragraph:

As a result of the above operations, no transfer over the USB bus is authorized when the flag is in the logic 1 state. There is, therefore, a dependency between the time for processing an interruption and the time delay in accepting the following transfer, the time for processing the interruption being linked to the microcontroller's operating frequency. Further, the time delay between each transaction depends on the master apparatus in that if that time delay is shorter than the minimum time for processing an interruption by the microcontroller, the following transfer cannot be authorized. This can result in the failure of the transaction.

Please replace the paragraph beginning at page 11, line 27, with the following rewritten paragraph:

By virtue of the invention, the first STATE 5 to STATE 0 transition generates a interruption CTR of the microcontroller, while the second STATE 0 to STATE 1 transition generates an interruption SOVR. These interruptions are processed sequentially by a program according to the flow chart of ~~FIG.5~~ FIG. 5. However, the information concerning the interruption type CTR or SOVR is first stored in a variable designated USB#1 Event. If this variable already includes information concerning a previous USB event, the new information is stored in a second USB#2 Event variable. The USB#1 and USB#2 Event variables are managed by the main loop of the program according to the flow chart of ~~FIG.6~~ FIG. 6.

Please replace the paragraph beginning at page 12, line 6, with the following rewritten paragraph:

The USB interruption processing routine includes a step 120 of starting USB interruption processing, and a step 122 of detecting the type of CTR interruption for the peripheral considered. If a positive response is provided, a step 124 of determining if the USB#1 Event variable already contains ~~an~~ a USB event is performed. If a negative response is provided, a step 126 is performed to place the information concerning the interruption in the USB#1 Event variable. The routine then

4

terminates by an end of USB interruption processing at step 130.

5